TU7 Agrobot

README for Code

There are 3 main folders which have the code

1. **TroughBotCode**
    - This folder has a Keil project for the Trough Side Code
    - To run this code open the **Serial_Comm_Xbee.uvproj** uvision project in keil and build the project.
    - Transfer the **Serial_Comm_Xbee.bin** file generated in the folder
    - Trough Bot will now send the water levels at regular intervals

2. **CentralServerCode**
    - This folder has a project which needs to be opened in Netbeans (or Eclipse).
    - A **.jar** file and a **.dll** file are also present which contains the library needed for XBee communication.
    - Add the given jar file to the project path.
    - Add the following line to the VM options in the project properties
      -Djava.library.path=PATH_TO_THE_DLL_FILE

    There are 3 main files:
    - **Communicator.java**
        - This file contains the code for opening and initialising serial ports and input-output streams for communication between troughs, server and the bot.
    - **Server.java**
        - This file is the main file in the project that needs to be run on the server side.
        - The server listens to the requests from the troughs and passes it to the bot which then waters the troughs.
        - It also displays a GUI pane which shows the water level status in the GUI. It is dynamically updated as the water level changes
    - **app.java**
        - This file has the code for the GUI.
        - Class app contains function *createAndShowGUI* which takes no. of troughs and labels(names) to troughs and displays a User Interface showing water levels of each trough in form of a progress bar.
        - Class app contains function *updateTroughs* which takes water levels of each trough in form of int array and updates water levels in User Interface.

To resolve the dependency, add the **RXTXComm.jar** file and the path to the **rxtxSerial.dll** file to the project (Note that this file is for 32-bit OS).
Set the Server.java class as the main class and run the file.
The GUI Pane is displayed and the server waits for the trough requests.

3. **WateringBotCode**
    - This folder has a Keil project for the Watering Bot.
    - To run this code open the **Serial_Comm_Xbee.uvproj** uvision project in keil and build the project.
    - Transfer the **Serial_Comm_Xbee.bin** file generated to the watering bot.
    - Watering bot will now respond to the server requests.

Bug Report

- There is a minor bug in the code. The **writeData()** function in **Server.java** file sends the byte to the bot in an **encoded** format because of which the Bot is not able to recognize the byte that is sent. The normal communication works completely fine with troughs sending the message to the server, which interprets the request and breaks it into trough wise request and sends it to the watering bot. If proper bytes are sent to the WateringBot, it responds to the requests properly (this can be seen using X-CTU terminal and XBees)

- The line follower code in the WateringBot has to be modified as it depends on the motors present in the Bot. In the current code, it is assumed that right motor is slow as compared to left motor, thats why a difference in PWM is maintained while moving.